# UNEM: UNrolled Generalized EM for Transductive Few-Shot Learning

## Supplementary Material

## A. Details on the minimization steps of the GEM optimization algorithm

The optimization algorithm alternates between a minimization step w.r.t. the distribution parameters and one w.r.t. the assignment variables. In the following, $\ell$ designates the current iteration.

**• Minimization step w.r.t. the distribution parameter**

For every $k \in \{1, \dots, K\}$, the first estimation step w.r.t. $\boldsymbol{\theta}_k$, with $\boldsymbol{u}_n = (u_{n,k}^{(\ell)})_{1 \leq k \leq K}$ given, is performed by considering the following optimization problem:

$$\underset{\boldsymbol{\theta}_k}{\text{minimize}} \quad -\sum_{n=1}^{N} u_{n,k}^{(\ell)} \ln \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k\right), \tag{12}$$

For a pdf belonging to the exponential family, this optimization problem is a convex. For instance, in the case of a Gaussian distribution whose pdf is defined in (5), the negative log-likelihood term, designated by function $F$, reduces to

$$F(\boldsymbol{\theta}_k) = \frac{1}{2} \sum_{n=1}^{N} u_{n,k}^{(\ell)} \|\boldsymbol{z}_n - \boldsymbol{\theta}_k\|^2. \tag{13}$$

The minimization of the above function (13) w.r.t $\boldsymbol{\theta}_k$ results in an explicit form of the estimated distribution parameter $\boldsymbol{\theta}_k^{(\ell+1)}$ given by

$$\boldsymbol{\theta}_k^{(\ell+1)} = \frac{\sum_{n=1}^{N} u_{n,k}^{(\ell)} \boldsymbol{z}_n}{\sum_{n=1}^{N} u_{n,k}^{(\ell)}}. \tag{14}$$

In turn, in the case of Dirichlet distribution whose pdf is defined in (6), the negative log-likelihood term reads

$$F(\boldsymbol{\theta}_k) = \sum_{n=1}^{N} u_{n,k}^{(\ell)} \Bigg( -\sum_{i=1}^{K} (\theta_{k,i} - 1) \ln z_{n,i} + \sum_{i=1}^{K} \ln \Gamma(\theta_{k,i}) - \ln \Gamma\left(\sum_{i=1}^{K} \theta_{k,i}\right) \Bigg). \tag{15}$$

Unlike the Gaussian model, the minimization of Dirichlet negative log-likelihood (15) has no closed form solution. To circumvent this problem, we resort to the Majorization-Minorization (MM) strategy recently developed in [33]. Thus, the estimated distribution parameter $\boldsymbol{\theta}_k^{(\ell+1)}$ can be expressed as follows

$$\boldsymbol{\theta}_k^{(\ell+1)} = \mathrm{MM}(\boldsymbol{u}_{\cdot,k}^{(\ell)}, \boldsymbol{\theta}_k^{(\ell)}). \tag{16}$$

**• Minimization step w.r.t. the assignment variable**

For every $n \in \mathbb{Q}$, the second estimation step w.r.t. $\boldsymbol{u}_n$ is achieved by minimizing the objective function (1), while keeping the distribution parameter set to the estimated vector $\boldsymbol{\theta}_k^{(\ell+1)}$. However, since the partition complexity term $\Psi$ is non convex, it is replaced by a linear tangent upper bound. More specifically, the following tangent inequality can be used:

$$\pi_k \ln \pi_k \geq \pi_k^{(\ell+1)} \ln \pi_k^{(\ell+1)} + (1 + \ln \pi_k^{(\ell+1)})(\pi_k - \pi_k^{(\ell+1)}) \tag{17}$$

Knowing that $\pi_k = \frac{1}{|\mathbb{Q}|} \sum_{n \in \mathbb{Q}} u_{n,k}$, the optimization problem (1) can be rewritten as follows

$$\underset{\boldsymbol{u}_n}{\text{minimize}} \quad G(\boldsymbol{u}_n) \tag{18}$$

with

$$\begin{aligned} G(\boldsymbol{u}_n) = &-\sum_{k=1}^{K} u_{n,k} \ln \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k^{(\ell+1)}\right) \\ &- \lambda \sum_{k=1}^{K} \frac{(1 + \ln \pi_k^{(\ell+1)})}{|\mathbb{Q}|} (u_{n,k} - u_{n,k}^{(\ell)}) \\ &+ T \sum_{k=1}^{K} u_{n,k} \ln u_{n,k} + \gamma_n \left( \sum_{k=1}^{K} u_{n,k} - 1 \right) \end{aligned} \tag{19}$$

where $\gamma_n$ is a Lagrange multiplier aiming to enforce the sum-to-one constraint. The nonnegativity constraint can be dropped since we will show next that it is satisfied by the minimizer of $G$ subject to the sum-to-one constraint.

The above optimization problem is convex. By cancelling the derivative of the above objective function (19) w.r.t. $u_{n,k}$, it can be checked that

$$\ln u_{n,k} = -1 - \frac{\gamma_n}{T} + \frac{1}{T} \Bigg( \ln \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k^{(\ell+1)}\right) + \frac{\lambda}{|\mathbb{Q}|} (1 + \ln \pi_k^{(\ell+1)}) \Bigg). \tag{20}$$

By applying the exponential function to (20) and determining the multiplier $\gamma_n$ so that the sum-to-one constraint is satisfied, it can be deduced that the optimal class assignment vector $\boldsymbol{u}_n^{(\ell+1)}$ is obtained by applying the softmax function:

$$\boldsymbol{u}_n^{(\ell+1)} = \text{softmax}\left(\frac{1}{T}\left(\ln \text{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k^{(\ell+1)}\right) + \frac{\lambda}{|\mathbb{Q}|}\ln(\pi_k^{(\ell+1)})\right)_k\right). \tag{21}$$

## B. Generalized EM algorithm in the case of Gaussian distribution

### B.1. Feature representation in vision-only few-shot-setting

Let us consider a few-shot scenario for vision-only models. Thus, for all dataset samples $\boldsymbol{x}_n$ with $n \in \{1, \ldots, N\}$, the feature vectors $\boldsymbol{z}_n$ are generated using a visual feature extractor $f^{(v)}$ as follows

$$\boldsymbol{z}_n = T_z f^{(v)}(\boldsymbol{x}_n) \tag{22}$$

where $T_z$ is a positive scaling parameter.

### B.2. Optimization algorithm

Using (13), (14), and (21), the proposed GEM algorithm reduces to Algorithm 2 in the case of a Gaussian distribution model.

**Algorithm 2** GEM-Gaussian based few-shot classification algorithm

**Input:** Compute $\boldsymbol{z}_n$ for the dataset samples and, for all $k \in \{1, \ldots, K\}$, initialize $\boldsymbol{\theta}_k^{(0)}$ as the means computed on the support set, and $\pi_k^{(0)} = 1$
**for** $\ell = 0, 1, \ldots, L-1$ **do**
// Update assignment vectors for all query samples
$\boldsymbol{u}_n^{(\ell+1)}$
$= \text{softmax}\left(\frac{1}{T}\left(-\frac{1}{2}\|\boldsymbol{z}_n - \boldsymbol{\theta}_k^{(\ell)}\|^2 + \frac{\lambda}{|\mathbb{Q}|}\ln(\pi_k^{(\ell)})\right)_k\right)$
// Update the mean parameter for each class
$\boldsymbol{\theta}_k^{(\ell+1)} = \dfrac{\sum_{n=1}^N u_{n,k}^{(\ell+1)} \boldsymbol{z}_n}{\sum_{n=1}^N u_{n,k}^{(\ell+1)}}, \quad \forall k \in \{1, \ldots, K\},$
// Update class proportions
$\pi_k^{(\ell+1)} = \dfrac{1}{|\mathbb{Q}|}\sum_{n\in\mathbb{Q}} u_{n,k}^{(\ell+1)}, \quad \forall k \in \{1, \ldots, K\},$
**end for**

## C. Generalized EM algorithm in the case of Dirichlet distribution

### C.1. Feature representation in few-shot CLIP

Our second few-shot scenario is devoted to vision-language models such as CLIP. Let us assume $f^{(v)}$ a vision-based feature extractor, and $f^{(l)}$ a language-based feature extractor. Thus, for a sample $\boldsymbol{x}_n$ with $n \in \{1, \ldots, N\}$ and a text prompt $\boldsymbol{t}_k$ of class $k \in \{1, \ldots, K\}$ (for example $\boldsymbol{t}_k$ = "a photo of a {class $k$}"), the visual and text features are given by $f^{(v)}(\boldsymbol{x}_n)$ and $f^{(l)}(\boldsymbol{t}_k)$, respectively. Then, the resulting feature embeddings of the data sample $\boldsymbol{x}_n$ is defined as its probability vector of belonging to class $k$:

$$\boldsymbol{z}_n = \text{softmax}\left\{T_z \cos\left(f^{(v)}(\boldsymbol{x}_n), f^{(l)}(\boldsymbol{t}_k)\right)_{1\leq k\leq K}\right\}, \tag{23}$$

where $T_z > 0$ is a temperature scaling parameter.

### C.2. Optimization algorithm

Using (16) and (21), and in the case of a Dirichlet data distribution model, the proposed GEM algorithm yields Algorithm 3.

**Algorithm 3** GEM-Dirichlet based few-shot classification algorithm

**Input:** Compute $\boldsymbol{z}_n$ for the dataset samples, initialize $\boldsymbol{u}_n^{(0)} = \boldsymbol{z}_n$, and $\boldsymbol{\theta}_k^{(0)} = \mathbf{1}_K$
**for** $\ell = 0, 1, \ldots, L-1$ **do**
// Update the Dirichlet parameter for each class
$\boldsymbol{\theta}_k^{(\ell+1)} = \text{MM}(\boldsymbol{u}_{\cdot,k}^{(\ell)}, \boldsymbol{\theta}_k^{(\ell)}), \quad \forall k \in \{1, \ldots, K\},$
// Update class proportions
$\pi_k^{(\ell+1)} = \dfrac{1}{|\mathbb{Q}|}\sum_{n\in\mathbb{Q}} u_{n,k}^{(\ell)}, \quad \forall k \in \{1, \ldots, K\},$
// Update assignment vectors for all query samples
$\mathcal{L}_{n,k}^{(\ell)} = \sum_{i=1}^K (\theta_{k,i}^{(\ell+1)} - 1)\ln z_{n,i}$
$\quad - \sum_{i=1}^K \ln\Gamma(\theta_{k,i}^{(\ell+1)}) + \ln\Gamma\left(\sum_{i=1}^K \theta_{k,i}^{(\ell+1)}\right)$
$\boldsymbol{u}_n^{(\ell+1)}$
$= \text{softmax}\left(\frac{1}{T}\left(\mathcal{L}_{n,k}^{(\ell)} + \frac{\lambda}{|\mathbb{Q}|}\ln(\pi_k^{(\ell+1)})\right)_k\right)$
**end for**

## D. Additional results

### D.1. Ablation studies

In this part, we perform ablation studies to illustrate the impact of the network depth, the effects of the introduced temperature scaling parameter and the benefits of learning adaptive hyper-parameters across the unrolled architecture layers.

**• Impact of the unrolled architecture depth**
First, we propose to analyze the impact of the number $L$ of layers of our unrolled architecture on the model accuracy, model size, and computational time. Table 4 reports the results. Thus, one of the main advantages of our UNEM model is that a few layers (about 7 or 10) are enough to achieve good performance. In what follows, the

experiments are conducted using $L = 10$.

| #Layers ($L$) | #Params | Acc. | Train Time (s) | Inference Time/task (s) |
|---|---|---|---|---|
| 3 | 7 | 65.6 | 2.80 | $3.04 \times 10^-2$ |
| 5 | 11 | 65.9 | 3.22 | $3.09 \times 10^-2$ |
| 7 | 15 | 66.3 | 3.46 | $3.21 \times 10^-2$ |
| 10 | 21 | 66.4 | 3.61 | $3.36 \times 10^-2$ |
| 12 | 25 | 66.2 | 4.06 | $3.45 \times 10^-2$ |
| 15 | 31 | 66.1 | 4.29 | $3.48 \times 10^-2$ |
| 18 | 37 | 66.2 | 5.74 | $3.68 \times 10^-2$ |

Table 4. Impact of the number of layers ($L$) on UNEM-Gaussian performance using *mini*-ImageNet, 5-shot and ResNet18.

**• Effects of temperature scaling**

To perform this study, we compare our unrolled architectures (UNEM-Gaussian as well as UNEM-Dirichlet) in both cases: (i) without introducing the temperature scaling parameter (as considered in the original algorithms PADDLE [32] and EM-Dirichlet [33]); (ii) while incorporating the temperature scaling (as proposed in our GEM algorithm).

Tables 5 and 6 depict the accuracy results in vision-only few-shot setting. Thus, it can be noticed that including the temperature scaling yields an accuracy improvement, which may vary from 1% to 3%. Moreover, in the context of vision-language models whose accuracy results are shown in Table 7, similar gains (reaching up to 3%), depending on the target donwstream dataset, are also achieved. This confirms again the advantage of incorporating the temperature scaling in our generalized algorithm.

**• Fixed vs adaptive hyper-parameters across layers**

One of the key advantages of unrolling algorithms is their flexibility in optimizing hyper-parameters, while allowing them to vary across the architecture layers. To show the potential of such hyper-parameter optimization approach, we propose to compare the proposed unrolled architectures (UNEM-Gaussian and UNEM-Dirichlet) in the following two cases: (i) the hyper-parameters are set fixed across the layers (as it is generally considered in original iterative algorithms), (ii) a set of hyper-parameters, adapted to the different layers, is learned.

Tables 8 and 9 provide the accuracy results for fixed and adaptive hyper-parameters optimization with vision-only models. It can be seen that learning adaptive hyper-parameters yields an accuracy gain of about 2-4% compared to the case when the hyper-parameters are kept fixed across layers. Similar comparisons are also performed with vision-language models as shown in Table 10. In this context, the improvement achieved by learning adaptive hyper-parameters often ranges from 1 to 2%.

### D.2. Performance under distribution shifts

In Table 11, we include the accuracy of the original PADDLE algorithm [32] and its unrolled version, with *tiered*-ImageNet used for pre-training, and a fine-grained classification dataset (CUB) as well as *mini*-ImageNet used for inference. One could observe similar improvements (about 4-7%) brought by UNEM.

### D.3. Backbone effect in CLIP

The performance of EM-Dirichlet and UNEM-Dirichlet using ViT-B/32 as backbone are shown in Table 12. Thus, in comparison to CLIP with ResNet50 (see Table 3), one may observe a similar or slightly better accuracy performance with most datasets. Moreover, the gains brought by UNEM over the iterative variant are consistent with those observed with ResNet50 backbone.

## E. Illustration and analysis of the learned hyperparameters

In this part, we propose to illustrate the variations of the learned hyper-parameters and analyze their orders-of-magnitude.

**• Illustration of the learned hyper-parameters**

The evolutions of the learned hyper-parameters $\lambda^{(\ell)}$ and $T^{(\ell)}$ with respect to the layer index are illustrated in Figures 4 and 5 for some downstream image classification tasks. While Figure 4 shows that the learned hyper-parameters with CUB (ResNet18), *mini*-ImageNet (ResNet18), and *mini*-ImageNet (WRN28-10) have similar amplitudes, much different orders-of-magnitude are observed with vision-language models as shown in Figure 5 for some test datasets. Let us recall that the different learned hyper-parameters, for all datasets, are available at https://github.com/ZhouLong0/UNEM-Transductive.

**• Analysis of the learned hyper-parameters**

Different observations could be made from the previous illustrations. On the one hand, in the case of vision-only models, it can be seen that the learned hyper-parameters $\lambda^{(\ell)}$ appear quite similar. However, the evolution of $T^{(\ell)}$ values shows different behaviors. Moreover, it is important to note that the optimal hyper-parameters also depend on the pre-training model as observed with *mini*-ImageNet (ResNet18) and *mini*-ImageNet (WRN28-10). On the other hand, with vision-language models, it can be observed that both hyper-parameter values $\lambda^{(\ell)}$ and $T^{(\ell)}$ strongly depend on the target dataset. Indeed, unlike the vision-only models where the feature vectors have a fixed size (which is equal to the dimension of the pre-trained model's output), the feature vectors $\boldsymbol{z}_n$ in the context of few-shot CLIP have different sizes, depending on the number of classes of each target

| Temperature scaling | Backbone | *mini*-ImageNet ($K = 20$) | | | *tiered*-ImageNet ($K = 160$) | | |
|---|---|---|---|---|---|---|---|
| | | 5-shot | 10-shot | 20-shot | 5-shot | 10-shot | 20-shot |
| × | ResNet-18 | 66.1 | 75.4 | 80.3 | 49.7 | 63.2 | 70.0 |
| ✓ | | **66.4** | **75.6** | **80.4** | **52.3** | **65.7** | **73.2** |
| × | WRN28-10 | **71.9** | 78.9 | 82.8 | 52.0 | 65.8 | 73.0 |
| ✓ | | 71.6 | **79.2** | **83.7** | **54.1** | **66.8** | **74.7** |

Table 5. Effects of the temperature scaling on the accuracy performance of UNEM-Gaussian approach applied to *mini*-ImageNet and *tiered*-ImageNet datasets.

| Temperature scaling | CUB ($K = 50$) | | |
|---|---|---|---|
| | 5-shot | 10-shot | 20-shot |
| × | 78.1 | 85.2 | 88.6 |
| ✓ | **78.5** | **85.3** | **88.6** |

Table 6. Effects of the temperature scaling on the accuracy performance of UNEM-Gaussian approach applied to CUB dataset.

| Temperature scaling | Food101 | EuroSAT | DTD | OxfordPets | Flowers102 | Caltech101 | UCF101 | FGVC Aircraft | Stanford Cars | SUN397 |
|---|---|---|---|---|---|---|---|---|---|---|
| × | 90.6 | 51.9 | **65.4** | 95.4 | 92.0 | 92.4 | **79.1** | 27.5 | 78.2 | 88.4 |
| ✓ | **91.4** | **53.8** | 65.3 | **96.0** | **95.6** | **93.4** | 78.5 | **30.4** | **80.0** | **88.5** |

Table 7. Effects of the temperature scaling on the accuracy performance of UNEM-Dirichlet approach applied to the vision-language models.

| Params across the layers | Backbone | *mini*-ImageNet ($K = 20$) | | | *tiered*-ImageNet ($K = 160$) | | |
|---|---|---|---|---|---|---|---|
| | | 5-shot | 10-shot | 20-shot | 5-shot | 10-shot | 20-shot |
| Fixed | ResNet-18 | 62.5 | 72.5 | 78.0 | 49.8 | 63.6 | 70.4 |
| Adaptive | | **66.4** | **75.6** | **80.4** | **52.3** | **65.7** | **73.2** |
| Fixed | WRN28-10 | 68.7 | 77.0 | 82.0 | 51.6 | 64.6 | 72.1 |
| Adaptive | | **71.6** | **79.2** | **83.7** | **54.1** | **66.8** | **74.7** |

Table 8. Fixed vs adaptive hyper-parameters setting in the UNEM-Gaussian approach, using *mini*-ImageNet and *tiered*-ImageNet datasets.

| Params across layers | CUB ($K = 50$) | | |
|---|---|---|---|
| | 5-shot | 10-shot | 20-shot |
| Fixed | 75.2 | 82.9 | 87.1 |
| Adaptive | **78.5** | **85.3** | **88.6** |

Table 9. Fixed vs adaptive hyper-parameters setting in the UNEM-Gaussian approach, using CUB dataset.

dataset. For instance, knowing that EuroSAT, Flowers102 and Stanford Cars have 10, 102, and 196 classes, respectively; it can be observed that the smallest (resp. largest) values of $\lambda^{(\ell)}$ are obtained with EuroSAT (resp. Stanford Cars). These results are expected since, by increasing the dimension of $\boldsymbol{z}_n$, the magnitude of the log-likelihood term may increase, and so, a higher value of $\lambda^{(\ell)}$ is needed to mitigate the class-balance bias.

This study shows the dependence of the introduced hyper-parameters on the target downstream dataset as well as the pre-training model, and confirms the importance of optimizing hyper-parameters in both evaluation scenarios.

| **Params across layers** | Food101 | EuroSAT | DTD | OxfordPets | Flowers102 | Caltech101 | UCF101 | FGVC Aircraft | Stanford Cars | SUN397 |
|---|---|---|---|---|---|---|---|---|---|---|
| Fixed | 89.6 | 52.2 | 64.8 | 95.3 | 95.3 | 92.3 | **79.2** | **31.6** | 78.0 | 87.6 |
| Adaptive | **91.4** | **53.8** | **65.3** | **96.0** | **95.6** | **93.4** | 78.5 | 30.4 | **80.0** | **88.5** |

Table 10. Fixed vs adaptive hyper-parameters setting in the UNEM-Dirichlet approach, using the vision-language models.

| Method | CUB | *mini*-ImageNet |
|---|---|---|
| PADDLE [32] | 66.0 | 82.9 |
| **UNEM-Gaussian** | **72.9** | **87.0** |

Table 11. Cross-domain evaluation: Accuracy performance on *mini*-ImageNet and CUB using a model trained on *tiered*-ImageNet, with a 5-shot setting and a ResNet18 backbone.

| Method | Food101 | EuroSAT | DTD | OxfordPets | Flowers102 | Caltech101 | UCF101 | FGVC Aircraft | Stanford Cars | SUN397 |
|---|---|---|---|---|---|---|---|---|---|---|
| EM-Dirichlet [33] | 89.3 | 54.8 | 63.9 | 92.5 | 92.7 | 92.8 | 77.2 | 27.3 | 73.9 | 81.7 |
| **UNEM-Dirichlet** | **91.4** | **57.5** | **67.4** | **95.7** | **95.2** | **94.0** | **80.4** | **33.8** | **77.8** | **88.4** |

Table 12. Accuracy performance of iterative EM-Dirichlet [33] and our UNEM variant using CLIP ViT-B/32 for feature extraction.

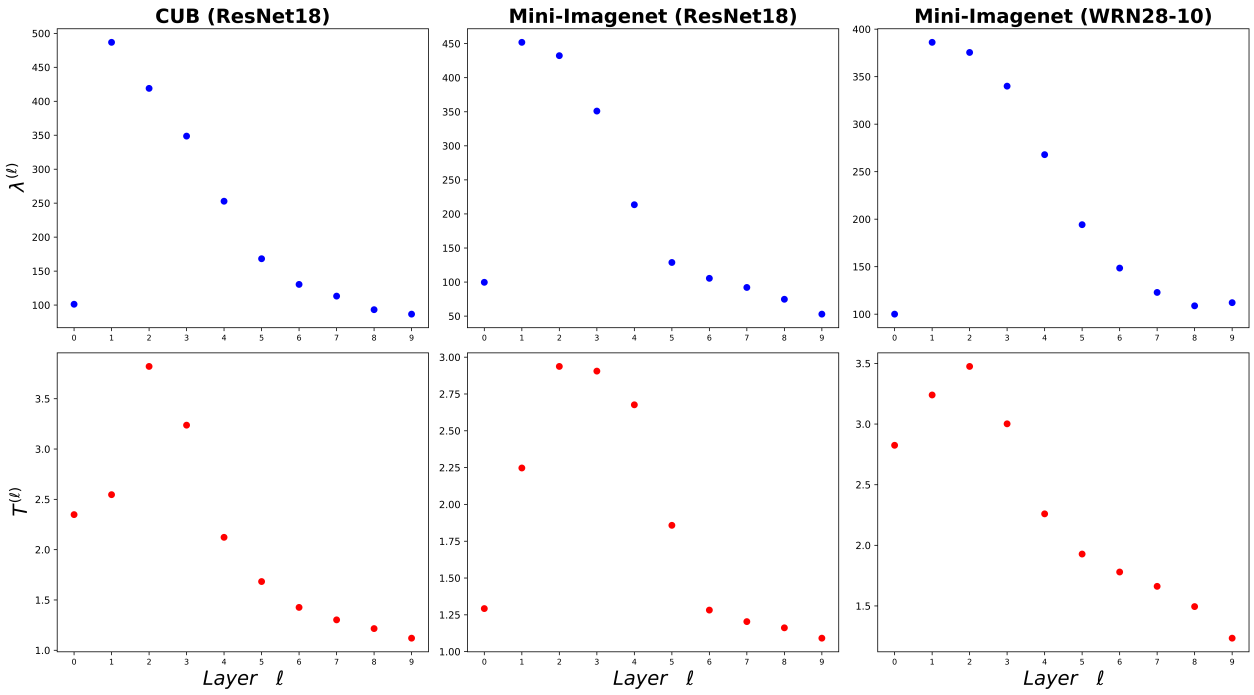


Figure 4. Illustration of the learned hyper-parameters $\lambda^{(\ell)}$ and $T^{(\ell)}$ across layers for CUB (with ResNet18 model), *mini*-ImageNet (with ResNet18 model) and *mini*-ImageNet (with WRN28-10 model).

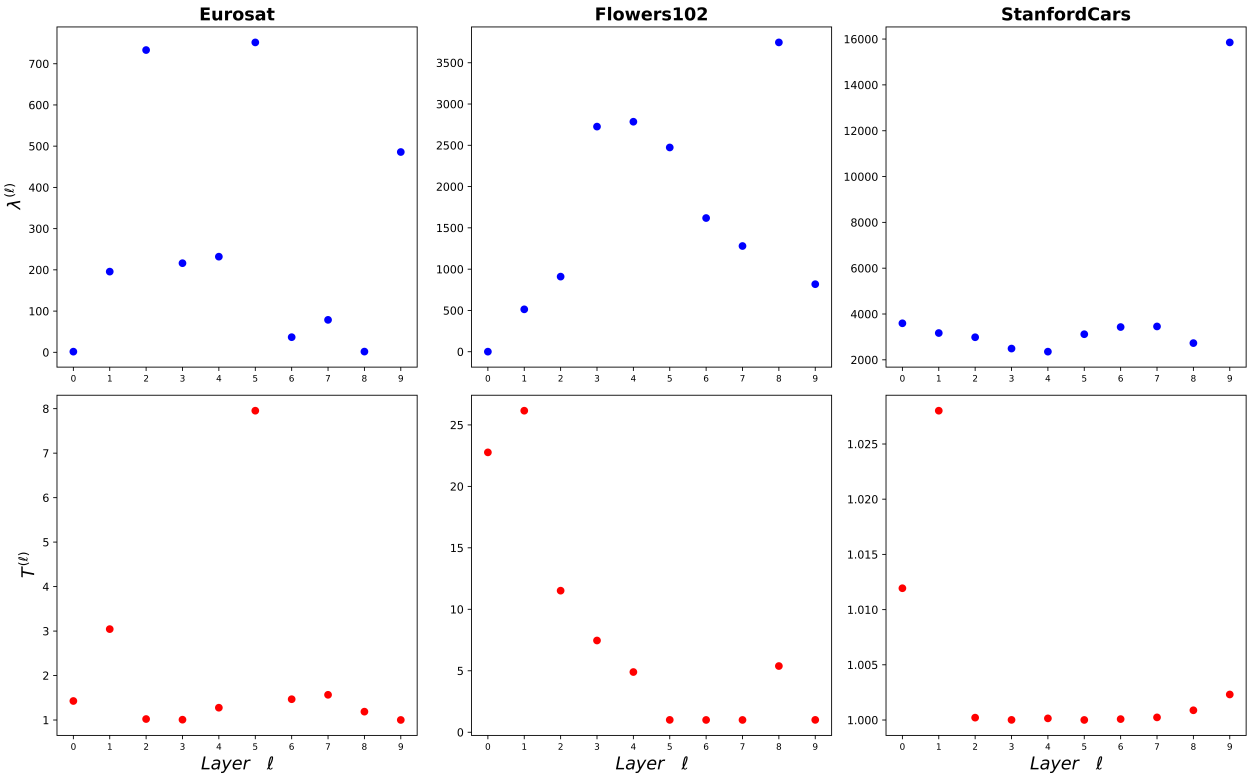


Figure 5. Illustration of the learned hyper-parameters $\lambda^{(\ell)}$ and $T^{(\ell)}$ across layers for some datasets with vision-language models.